

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-254742
(43)Date of publication of application : 25.09.1998

(51)Int.Cl. G06F 12/00
G06F 12/00
G06F 12/00

(21)Application number : 09-052774
(22)Date of filing : 07.03.1997

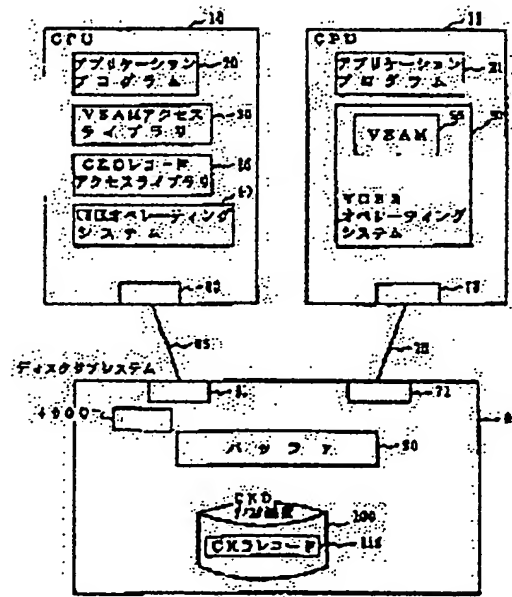
(71)Applicant : HITACHI LTD
(72)Inventor : KANDA MOTOHIRO
YAMAMOTO AKIRA
NAKANO TOSHIO
YOSHIDA MINORU

(54) COMPUTER SYSTEM

(57)Abstract

PROBLEM TO BE SOLVED: To share data between a main frame and an option system.

SOLUTION: A disk controller 80 is connected to CPU 10 which a UNIX operating system 40 controls with an SCSI(small computer system interface) interface 61 and is connected to CPU 11 which a VOS(virtual storage operating system) 3 operating system 50 controls with a channel interface 71. CPU 10 is provided with a CKD(count key data) record access library 35 and a VSAM (virtual storage access method) access library 30. CPU 10 accesses to a VSAM record which CPU 11 stores in the disk controller 80 by a CKD system by an FBA(fact block architecture) system and can accesses to the application program 20 of CPU 10 as the VSAM record with control information of VSAM as a base.



LEGAL STATUS

[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-254742

(43) 公開日 平成10年(1998) 9月25日

(51) Int.Cl.⁶
G 0 6 F 12/00

識別記号
5 1 1
5 1 4
5 2 0

F I
G 0 6 F 12/00

5 1 1
5 1 4 Z
5 2 0 A

審査請求 未請求 請求項の数 8 O L (全 17 頁)

(21) 出願番号 特願平9-52774

(22) 出願日 平成9年(1997) 3月7日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 神田 基博

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72) 発明者 山本 彰

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72) 発明者 中野 俊夫

神奈川県小田原市国府津2880番地 株式会社日立製作所ストレージシステム事業部内

(74) 代理人 弁理士 小川 勝男

最終頁に続く

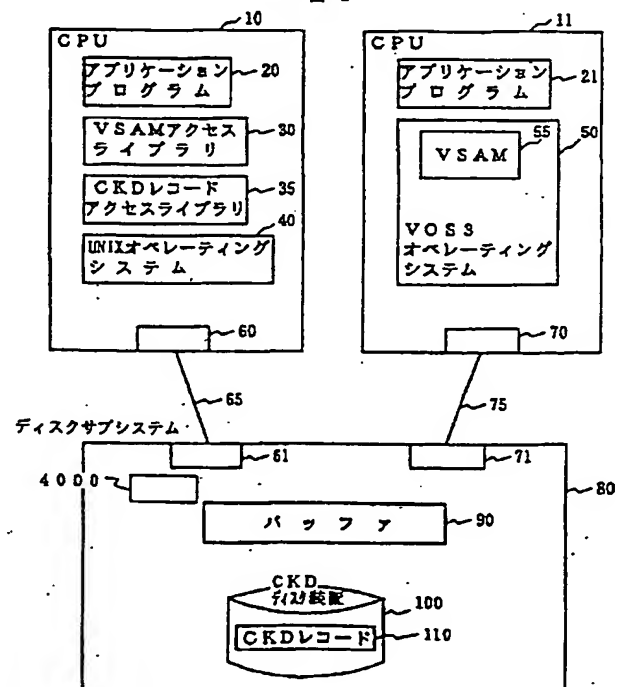
(54) 【発明の名称】 計算機システム

(57) 【要約】

【課題】 メインフレームと、オープンシステムとの間で、データの共用を可能とする。

【解決手段】 ディスク制御装置 80は、SCSIインタフェース 61でUNIXオペレーティングシステム 40が制御するCPU 10と接続し、チャンネルインタフェース 71で、VOS3オペレーティングシステム 50が制御するCPU 11と接続する。CPU 10には、CKDレコードアクセスライブラリ 35と、VSAMアクセスライブラリ 30があって、CPU 11がディスク制御装置 80にCKD形式で格納したVSAMレコードを、FBA形式でアクセスし、かつVSAMの制御情報を元にして、VSAMレコードとしてCPU 10のアプリケーションプログラム 20にアクセスできる。

図 1



【特許請求の範囲】

【請求項1】 カウントキーデータ形式に従うインタフェースと固定長ブロック形式に従うインタフェースとを有する記憶装置を含む計算機システムにおいて、シリンダー番号、ヘッド番号そしてレコード番号で指定されるカウントキーデータ形式のレコードアドレスと、論理ブロックアドレスで指定される固定長ブロック形式のアドレスとを互いに変換する手段と、カウントキーデータ形式で記憶装置に格納されているレコードを固定長ブロック形式に従うインタフェースでアクセスさせる手段と、前記レコードから、ユーザデータだけを取り出す手段と、前記レコードのユーザデータをあらかじめ定められた形式にしたがって解釈し、利用する手段とを有することを特徴とする計算機システム。

【請求項2】 ディスクアクセスをアクセスするディスク装置のカウントキーデータ形式にしたがって行う第1の計算機と、ディスクアクセスを固定長ブロック形式にしたがって行う第2の計算機と、前記第1及び第2の計算機に接続するディスクサブシステムとを含んで構成される計算機システムであって、前記ディスクサブシステムは、前記第1の計算機と接続し、カウントキーデータ形式に従う第1のインタフェースと、前記第2の計算機と接続し、固定長ブロック形式に従う第2のインタフェースと、前記第1及び第2の計算機からアクセスされるデータをカウントキーデータ形式で格納するディスク装置と、前記第2のインタフェースから受けた固定長ブロック形式のアドレスに基づき前記ディスク装置からデータを読み出し、該読み出したデータを前記第2のインタフェースを介して固定長形式のブロック単位で前記第2の計算機に転送する手段を有し、前記第2の計算機は、アプリケーションプログラムから発行されるカウントキーデータ形式に従ったディスクアクセス要求を受け付け、カウントキーデータ形式のアドレスを前記固定長ブロック形式のアドレスに変換する手段と、変換された固定長ブロック形式のアドレスを用いて前記ディスクサブシステムへアクセスする手段と、前記ディスクサブシステムから転送されたブロック単位のデータからアクセス要求で指定されたレコードのデータを抽出する手段を有することを特徴とする計算機システム。

【請求項3】 前記転送手段は、前記固定長形式のアドレスに基づき前記アクセス要求で指定されるレコードを格納した領域のシリンダー番号、ヘッド番号を算出して前記指定レコードを含むトラックのデータを前記ディスク装置から読み出し、前記固定長形式のアドレスにより定まる前記トラック内でのブロックの先頭位置を判定し、前記トラック内の前記先頭位置以降にあるデータを前記固定長形式に従ったブロック単位で前記第2の計算機に転送することを特徴とする請求項2記載の計算機システム。

【請求項4】 前記転送手段は、転送中のデータブロック

の途中で前記トラックのデータ転送を終了した場合、該転送中のデータブロックの終端まで予め定められたデータを転送することを特徴とする請求項3記載の計算機システム。

【請求項5】 前記ディスク装置は、固定長の記憶領域を有し、前記カウントキーデータ形式のデータを前記固定長の記憶領域の大きさに分割して格納しており、前記転送手段は、前記固定長形式のアドレスに基づき、転送すべきデータを格納した前記ディスク装置の記憶領域を判定してデータの読み出しを行うことを特徴とする請求項2記載の計算機システム。

【請求項6】 前記転送手段は、前記固定長形式のアドレスに基づいて、前記カウントキーデータ形式のデータをアクセスするためのシリンダー番号、ヘッド番号、及びレコード番号を算出し、算出したシリンダー番号、ヘッド番号、及びレコード番号から転送すべきデータを格納した記憶領域のアドレスを算出することを特徴とする請求項5記載の計算機システム。

【請求項7】 カウントキーデータ形式に従うインタフェースと、固定長ブロック形式に従うインタフェースの両方を有する記憶装置を含む計算機システムにおいて、シリンダー番号、ヘッド番号そしてレコード番号で指定されるカウントキーデータ形式のレコードアドレスと、LBA (Logical Block Address) で指定される固定長ブロック形式のアドレスとを互いに変換する手段と、カウントキーデータ形式で記憶装置に格納されているレコードを固定長ブロック形式に従うインタフェースでアクセスさせる手段と、前記レコードから、ユーザデータだけを取り出す手段と、前記レコードのユーザデータを、あらかじめ定められた形式にしたがって解釈し、利用する手段とを有することを特徴とする計算機システム。

【請求項8】 請求項7記載の計算機システムであって、カウントキーデータ形式に従うインタフェースでデータをアクセスし、当該データを管理する第1の管理手段と、

固定長ブロック形式に従うインタフェースでデータをアクセスし、当該データを管理する第2の管理手段と、前記第1及び第2の管理手段のいずれかがあるデータを使用している場合、もう片方の手段にとっても当該データが使用中であると、もう片方の手段をして知らしめる手段とを有することを特徴とする計算機システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、計算機システムに関し、さらに詳しくは、カウントキーデータ形式に従うインタフェースと、固定長ブロック形式に従うインタフェースの両方を有する記憶装置の使用法に関する。

【0002】

【従来の技術】異なるオペレーティングシステム間のファイル共有あるいはファイル変換については、従来の技術で公知のものがある。例えば、日本サンマイクロシステムズ株式会社発行の、JLEリファレンスマニュアル3の、装置とネットワークインタフェースのPCFS(4S)には、UNIXオペレーティングシステムから、MS-DOSオペレーティングシステムのファイルをアクセスする機能が記載されている。(JLEは、日本サン・マイクロシステムズ株式会社の商標です。UNIXは、米国X/Open Company Ltd. の米国及びその他の国における登録商標です。MS-DOSは、米国Microsoft corporation の米国及びその他の国における登録商標です。)

【0003】

【発明が解決しようとする課題】近年、パソコンやワークステーションなどのいわゆるオープンシステムを使用して、従来メインフレームで行なわれてきた業務を行なう、いわゆるダウンサイジングが盛んに行なわれている。

【0004】メインフレームは伝統的にディスクアクセスをカウントキーデータ形式に従って行なってきたのに比べて、オープンシステムはディスクアクセスを固定長ブロック形式に従って行なう。このために、一般的にはメインフレームで使用しているディスクは、オープンシステムでは使用できないという問題がある。メインフレームで使用しているディスクには、すでに大量の業務上の情報が蓄積されているために、ダウンサイジングを行なっても、オープンシステムからこのディスクをアクセスしたいという顧客の要求は大きい。この要求に答えるための技術として、分散データベースや、ファイル転送があるが、ネットワーク負荷が高くなるとか、既存の業務プログラムの変更が必要になるなどの欠点がある。

【0005】本発明は、ディスクにカウントキーデータ形式に従うインタフェースと、固定長ブロック形式に従うインタフェースの両方を持たせて、オープンシステムからこのディスクをアクセスし、メインフレームから格納した業務上の情報を利用することを可能とするものである。

【0006】カウントキーデータ形式に従うインタフェースと、固定長ブロック形式に従うインタフェースの両方を有する記憶装置の使用法、特に2つのインタフェースの間でデータを共有する機能については、従来の技術で公知のものはない。

【0007】

【課題を解決するための手段】本発明は、カウントキーデータ形式に従うインタフェースと、固定長ブロック形式に従うインタフェースの両方を有する記憶装置を含む計算機システムにおいて、シリンダー番号、ヘッド番号そしてレコード番号で指定されるカウントキーデータ形式のレコードアドレスと、LBA(Logical Block Address)で指定される固定長ブロック形式のアドレスとを

互いに変換する手段と、カウントキーデータ形式で記憶装置に格納されているレコードを固定長ブロック形式に従うインタフェースでアクセスさせる手段と、前記レコードから、ユーザデータだけを取り出す手段と、前記レコードのユーザデータを、あらかじめ定められた形式にしたがって解釈し、利用する手段とを有する計算機システムを提供する。

【0008】さらに、本発明は、カウントキーデータ形式に従うインタフェースでデータをアクセスし、当該データを管理する手段と、固定長ブロック形式に従うインタフェースでデータをアクセスし、当該データを管理する手段と、前記どちらかの手段があるデータを使用している場合、もう片方の手段にとっても当該データが使用中であるともう片方の手段をして知らせる手段を有する計算機システムを提供する。

【0009】

【発明の実施の形態】以下、図に示す発明の実施の形態によりこの発明をさらに詳しく説明する。

【0010】一第1の発明の実施の形態一

図1は、本発明の第1の発明の実施の形態の計算機システムの要部構成図である。

【0011】CPU10は、SCSI(Small Computer System Interface)インタフェース60を有する。CPU11は、チャンネルインタフェース70を有する。ディスクサブシステム80は、SCSIインタフェース61でCPU10と接続し、チャンネルインタフェース71でCPU11と接続する。SCSIバス65は、SCSIインタフェース60と61をむすぶ。チャンネルケーブル75は、チャンネルインタフェース70と71をむすぶ。

【0012】SCSIインタフェース60ないし61を、任意の固定長ブロック形式に従うインタフェースとする構成も可能である。チャンネルインタフェース70ないし71を、任意のカウントキーデータ形式に従うインタフェースとする構成も可能である。

【0013】以下、カウントキーデータ形式をCKD形式と呼び、固定長ブロック形式をFBA(Fixed Block Architecture)形式と呼ぶ。CKD形式のレコードを、CKDレコードと呼ぶ。

【0014】CPU10は、UNIXオペレーティングシステム40によって制御される。CPU11は、日立製作所のVOS3(Virtual-storage Operating System 3)オペレーティングシステム50によって制御される。UNIXオペレーティングシステム40を、SCSIインタフェースをサポートする任意のオペレーティングシステムとした構成も可能である。VOS3オペレーティングシステム50を、チャンネルインタフェースをサポートする任意のオペレーティングシステムとした構成も可能である。

【0015】CPU10とCPU11では、アプリケーションプログラム20と21がそれぞれ動作する。アプ

リケーションプログラム20と21は、プログラミング言語COBOLで記述され、ディスクサブシステム80にある、VSAM ESDS (Virtual Storage Access Method Entry-Sequenced Data Set) をアクセスする。VSAM ESDSを、CKD形式で格納されたデータをアクセスする、任意のアクセス法の任意のデータセットあるいは任意のファイルとする構成も可能である。VOS3オペレーティングシステム50は、VSAM55を含む。VSAM55は、ディスクサブシステム80に格納されたデータを、VSAMデータセットのレコードとして、アプリケーションプログラム21からアクセス可能とする。

【0016】図2は、UNIXオペレーティングシステム40の要部構成図である。

【0017】UNIXオペレーティングシステム40は、ファイル管理機能150およびSCSIデバイスドライバ155を含む。メモリ管理機能やプロセス管理機能など、本発明の実施の形態に関連のうすいものは省略してある。SCSIデバイスドライバ155は、SCSIインタフェース60を制御し、ディスクサブシステム80および180をアクセスする。ディスクサブシステム80および180は、SCSIバス65によりSCSIインタフェース60に接続する。ディスクサブシステム80は、VSAMデータセットを、ディスクサブシステム180はFFS (Berkeley Fast File System) のファイルを格納する。FFSは、UNIXオペレーティングシステム40のファイル管理機能150によって標準的にサポートされるファイルシステムの形式である。一方、UNIXオペレーティングシステム40のファイル管理機能150は、VSAMデータセットのアクセスをサポートしていないものとする。

【0018】lsコマンド160は、指定したディレクトリに存在するファイルの一覧を表示するコマンドである。ディスクサブシステム180に含まれるFFSのあるディレクトリを指定して発行された、ls コマンド160のディスクアクセス要求は、ファイル管理機能150およびSCSIデバイスドライバ155によって処理されて、ディスクサブシステム180をアクセスする。

【0019】一方、CKDレコードアクセスライブラリ35のディスクアクセス要求は、ファイル管理機能150を経由せずに、SCSIデバイスドライバ155だけによって処理され、ディスクサブシステム80をアクセスする。この場合、ディスクサブシステム80に格納される各VSAMデータセットの属性情報や、物理的な格納位置情報等は、UNIXオペレーティングシステム40からCKDレコードアクセスライブラリ35へは提供されない。CKDレコードアクセスライブラリ35には、ディスク制御装置80全体が1つのファイルに見えるだけである。このようなアクセス方法を、ローIOと呼ぶ。

【0020】UNIXオペレーティングシステム40は、ファイル名により、前記2つのディスクアクセス要

求を区別する。アクセスするファイル名として、キャラクタ特殊ファイル名を指定すると、ローIOによるアクセスがされ、それ以外の通常ファイル名を指定すると、ファイル管理機能150を経由するアクセスとなる。キャラクタ特殊ファイル名の例として、/dev/rsd1c がある。

【0021】図1に戻って、CPU10には、VSAMアクセスライブラリ30と、CKDレコードアクセスライブラリ35がある。CKDレコードアクセスライブラリ35は、VSAMアクセスライブラリ30が発行する、CKDレコードアクセス要求を、FBA形式のアクセス要求に変換して、UNIXオペレーティングシステム40に与え、その結果をCKD形式に変換してVSAMアクセスライブラリ30に返す。VSAMアクセスライブラリ30は、ディスクサブシステム80に格納された、VSAMを制御するためのデータ構造を参照して、ディスクサブシステム80に格納されたデータを、VSAMデータセットのレコードとして、アプリケーションプログラム20からアクセス可能とする。

【0022】なお、UNIXオペレーティングシステム40のファイル管理機能150が、VSAMデータセットのアクセスをサポートする構成も可能である。この場合、VSAMアクセスライブラリ30およびCKDレコードアクセスライブラリ35はUNIXオペレーティングシステムの一部として実装される。

【0023】ディスクサブシステム80は、バッファ90とCKDディスク装置100を有する。CKDディスク装置100には、CKDレコード110が格納されている。ディスクサブシステム80は、チャネルインタフェース71を経由して、CKD形式にしたがった、シリンダー番号、ヘッド番号、そしてレコード番号を指定したレコードのアクセスを提供する。以下、シリンダー番号、ヘッド番号、そしてレコード番号で表されるレコードアドレスを、CCHHRと呼ぶ。シリンダー番号、ヘッド番号で表されるトラックアドレスを、CCHHと呼ぶ。

【0024】さらに、ディスクサブシステム80は、CKD-FBAレコード形式変換R/W処理A4000を使って、SCSIインタフェース61を経由して、FBA形式に従ったアクセスも提供する。CKD-FBAレコード形式変換R/W処理A4000は、SCSIインタフェース61を経由してFBAレコードを受け取り、レコードの格納されるCCHHRを計算し、レコード形式を、FBAから、CKDに変換し、そしてCKDレコードを格納する。逆の処理、つまりレコードの読み出しも同様に動作する。

【0025】FBAレコードのブロック長は512バイトとし、シリンダー0、ヘッド0、そしてレコード1を第0のLBA (Logical Block Address) をもつものとする。ディスクサブシステム80に格納されるCKDレコードの、FBA形式に従ったアドレス付けや、アクセスの

仕方については後述する。

【0026】CKDディスク装置100に格納されたデータは、当該データが格納されているシリンダー番号、ヘッド番号、そしてトラック先頭からのバイト位置によって識別される。データのCKDディスク装置100でのシリンダー番号、ヘッド番号は、当該データが、CPU11からチャネルインタフェース71を経由してアクセスされるときに指定されるシリンダー番号、ヘッド番号と同じであるとする。これが異なる構成も可能である。CKDディスク装置100に格納されたデータが、LBAで識別される構成も可能である。その場合、CPU11からチャネルインタフェース71を経由してアクセスされるときに指定されるシリンダー番号、ヘッド番号そしてレコード番号と、LBAを互いに変換する手段が必要になる。この構成は、第2の発明の実施の形態として後述する。

【0027】CKDディスク装置100が複数ある構成も可能である。

【0028】CKDディスク装置100は、1つのトラックあたり最大64キロバイトのデータを格納することができるものとし、以下これをトラック容量と呼ぶ。バッファ90は、トラック容量だけの大きさを持つ。トラック容量に、ヘッド数をかけたものを、シリンダー容量と呼ぶ。

【0029】図3は、アプリケーションプログラム20の処理のフローチャートである。

【0030】アプリケーションプログラム20は、COBOLで記述され、ディスクサブシステム80に格納される、VSAM ESDSを1つオープンして、データセットの最初のレコードを1つ読み、それをコンソールに表示し、データセットをクローズして終わるというものである。

【0031】ステップ1100で、データセットのオープンを行う。COBOLでの記述形式は、OPEN INPUT データセット名であり、指定したデータセットを読み込み専用でオープンすることを示す。これによりVSAMアクセスライブラリ30が呼ばれる。そこで行なわれる処理を、図4に示す。これについては後述する。

【0032】ステップ1110で、VSAMレコードの読み込みを行う。COBOLでの記述形式は、READ データセット名 RECORD INTO 一意名であり、指定したデータセットからVSAMレコードを1つ読んで、一意名で示されるアプリケーションプログラム20の変数にいれることを示す。これによりVSAMアクセスライブラリ30が呼ばれる。そこで行なわれる処理を、図5に示す。これについては後述する。

【0033】ステップ1120で、前記ステップ1110で読み込んだVSAMレコードを、コンソールに表示する。

【0034】ステップ1130で、データセットのクローズを行なう。これによりVSAMアクセスライブラリ

30が呼ばれる。そこで行なわれる処理を、図6に示す。これについては後述する。

【0035】以上でアプリケーションプログラム20の処理を終わる。

【0036】図4は、VSAMアクセスライブラリ30が行う、VSAM ESDSのオープン処理1100のフローチャートである。

【0037】ステップ500で、UNIXオペレーティングシステム40の、ローIO機能をオープンする。正確に言うと、ディスクサブシステム80に対応したキャラクタ特殊ファイルをオープンする。本発明の実施の形態では、すべてのVSAM ESDSは、ディスク制御装置80に存在するものとする。CPU10に複数のディスク装置が接続される構成も可能である。その場合は、指定されたデータセット名から、それがどのディスク装置に存在するかを判断する手段が必要である。

【0038】ステップ510で、VSAMアクセスライブラリ30は、CKDレコードアクセスライブラリ35の機能を使用して、ディスクサブシステム80から、標準ボリュームラベルを読み込む。

【0039】標準ボリュームラベルは、ディスク装置の特定の場所に書かれており、VTOC (Volume Table of Content) のアドレスを有する。標準ボリュームラベルの読み込みは、前記特定のCCHHRを指定して、図7に示すCKDレコード読み込み処理を呼ぶことで行う。CKDレコード読み込み処理については後述する。VOS3オペレーティングシステム50は、ディスク装置ごとに1つVTOCを作成する。VTOCは、そのディスク装置に含まれるすべてのデータセットの管理情報を有する。

【0040】ステップ510で、標準ボリュームラベルのデータ部から、VTOCのCCHHRを得る。

【0041】VTOCは、キー部44バイト、データ部96バイトのレコードの集まりである。各レコードを、DSCB (Dataset Control Block) と呼ぶ。DSCBには、いろいろな形式がある。形式1 DSCBは、キー部がデータセット名で、データ部はそのデータセットの属性や、割り当てられている物理的な格納位置の情報を有する。データセットが割り当てられている物理的な格納領域を、エクステンと呼ぶ。エクステンは、それが占めるシリンダー番号、トラック番号で識別される。

【0042】ステップ520で、VTOCの最初のレコードをCKDレコードアクセスライブラリ35の機能を使用して、1つ読む。

【0043】ステップ530で、これが形式1 DSCBであり、かつキー部は指定されたデータセット名に等しいか調べる。条件が成り立てば、ステップ550に進む。そうでなければ、ステップ540に進む。

【0044】ステップ550では、これが目的とするデータセットの形式1 DSCBであるので、これをVSAMアクセスライブラリ30のローカルな変数にコピーして、

コール元に戻る。これで、VSAM ESDSのオープン処理が終わる。

【0045】ステップ540では、レコードアドレスを進め、次のDSCBを読む準備をして、ステップ520に戻る。

【0046】図5は、VSAMアクセスライブラリ30が行う、VSAM ESDSのリード処理1110のフローチャートである。

【0047】VSAM ESDSに含まれるすべてのVSAMレコードは、それらが作成された順で順序づけされている。VSAMアクセスライブラリ30は現在処理対象となるVSAMレコード位置を常に記憶しており、以下、それをカレントレコードと呼ぶ。カレントレコードは、データセットがオープンされたときに、最初のVSAMレコードとされる。リードされると、カレントレコードは次のVSAMレコードとなる。

【0048】VSAMでは、ディスク装置との転送の単位を、CI(Control Interval)と呼んでいる。CIは、VSAMレコードを含むほか、CI内の未使用スペースの管理情報などを有する。VSAMアクセスライブラリ30は、VSAMレコードを、それが含まれるCIのRBA(Relative Byte Address)と、CIの中でのVSAMレコードの通番で識別する。

【0049】ステップ600で、目的RBAを、カレントレコードを含むCIのRBAとする。

【0050】ステップ610で、目的RBAを含むエクステントを探す。これはつまり、データセット内のオフセットから、そのオフセットを持つデータが実際に格納されているCCHHRを求めることである。

【0051】VSAM ESDSのオープンの直後のリード処理を例に、以下説明する。このときカレントレコードはVSAM ESDSの最初のCIに含まれ、そのCIのRBAは0であるとする。形式1 DSCBは、エクステント情報の配列を持ち、各エクステントごとに、そのエクステントの開始CCHHと、終了CCHHを記録する。前記エクステント情報の配列は、対応するRBAの昇順に並んでいる。今の例では、目的RBAは0であるので、明らかに最初のエクステントの最初のレコードに含まれる。この結果、目的とするCCHHは形式1 DSCBに記録された最初のエクステント情報の示す開始CCHHであることがわかる。また、レコード0にはユーザデータは格納できないため、Rは1であることがわかる。

【0052】ステップ620では、前記ステップ610で求めたCCHHRを指定して、図7に示すCKDレコード読み込み処理を呼ぶ。CIが、複数のCKDレコードからなる場合には、その数だけCKDレコード読み込み処理を呼ぶ。

【0053】ステップ630では、CI内の制御情報を参照して、CIの中で目的とするVSAMレコードを探す。それは、カレントレコードの通番を持つものである。

【0054】ステップ640で、アプリケーションプログラム20の変数である一意名に、VSAMレコードをコピ

ーする。

【0055】ステップ650で、カレントレコードを、次のVSAMレコードとする。

【0056】以上で、VSAM ESDSのリード処理が終わる。

【0057】図6は、VSAMアクセスライブラリ30が行う、VSAM ESDSのクローズ処理1130のフローチャートである。

【0058】ステップ1000で、UNIXオペレーティングシステム40の、ローIO機能をクローズする。正確に言うと、ディスクサブシステム80に対応したキャラクタ特殊ファイルをクローズする。

【0059】ステップ1010で、作業領域を解放する。

【0060】以上で、VSAM ESDSのクローズ処理が終わる。

【0061】図7は、CKDレコードアクセスライブラリ35が行なう、CKDレコード読み込み処理のフローチャートである。CKDレコード読み込み処理は、VSAMアクセスライブラリ30から呼ばれ、指定されたCCHHRを持つレコードをディスク制御装置80から読み、そのキー部とデータ部を返す。

【0062】ステップ400では、指定されたCCHHRを持つレコードを含むトラックの先頭に対応するLBAを計算する。ただし、以下では、割算を示す/では、整数部分のみを商とする。

【0063】
$$LBA = (CC * \text{シリンダー容量} + HH * \text{トラック容量}) / \text{ブロック長}$$

ステップ410では、UNIXオペレーティングシステム40のローIO機能をつかって、前記LBAから、1トラック長だけ、SCSIインタフェース60を経由してディスクサブシステム80からデータを読む。ローIO機能を使う理由は、UNIXオペレーティングシステム40はVSAMデータセットのアクセスをサポートしていないために、UNIXオペレーティングシステム40のファイル管理機能を使用することができないためである。

【0064】ステップ420では、ステップ430で処理するレコードアドレスを、トラックの先頭とする。

【0065】ステップ430では、当該レコードのカウンタ部に含まれる、レコードIDを調べて、当該レコードが、指定されたCCHHRを持つものであるか調べる。指定されたCCHHRを持てばステップ440へ進む。そうでなければ、ステップ450へ進む。

【0066】ステップ440では、目的とするレコードが見つかったので、そのキー部と、データ部をコール元に返して処理を終了する。

【0067】ステップ450では、レコードアドレスに、現在のレコードのキー部とデータ部の長さを加えて、つぎに処理するレコードアドレスとして、ステップ430に戻る。レコードのキー部とデータ部の長さは、

そのレコードのカウンタ部に書いてある。

【0068】以上、図7に示した処理では、CKDレコード読み込み処理が1度呼ばれると必ず1トラックがディスクサブシステム80から読まれる。最近アクセスしたトラックのデータをキャッシュしておいて、同じトラックが要求されたらディスクサブシステム80をアクセスしないですます構成も可能である。

【0069】図8は、ディスクサブシステム80が行なう、SCSI READ処理のフローチャートである。この処理は、前記図7のCKDレコード読み込み処理のステップ410でUNIXオペレーティングシステム40のローIO機能が使用された結果、ディスク制御装置80のCKD-FBAレコード形式変換R/W処理A4000で行なわれるものである。

【0070】ステップ200では、ディスクサブシステム80のCKD-FBAレコード形式変換R/W処理A4000は、CPU10からSCSIインタフェース61を経由してSCSIのREADコマンドを受ける。

【0071】ステップ210では、受領したCDB(Command Descriptor Block)から、対象データのLBAを得、それをCKDディスク装置100のシリンダー番号とヘッド番号に変換する。

【0072】シリンダー番号 = (LBA * ブロック長) / シリンダー容量

ヘッド番号 = ((LBA * ブロック長) % シリンダー容量) / トラック容量

トラック先頭からのバイト位置 = (LBA * ブロック長) % トラック容量

%は、剰余を表す。即ち、整数m、nに対して、 $n \% m$ はnをmで割った余りを示す。

【0073】ステップ220では、指定されたシリンダー番号とヘッド番号をもつトラックを、バッファ90に読む。このときにバッファ90に読み込まれるトラックの形式を、図9に示す。

【0074】図9は、CKDディスク装置100に格納されるデータの、トラック上の配置を示したものである。305から、345は、トラック上の実際の物理的なデータの配置を示す。これに対して、380は、前記図8のステップ220でバッファ90に読み込まれるトラックの形式を示したものである。390、392そして394は、380に示すデータが、SCSIインタフェース61を経由して、CPU10に転送されるとき形式を示したものである。

【0075】305は、ホームアドレスであり、トラックの状態や、IDを示す。310はレコード0のカウンタ部であり、315はレコード0のデータ部である。カウンタ部は、各レコードの最初のフィールドであり、そのレコードの状態、位置、及び長さを示す。レコード0は、トラックの最初のレコードであり、ユーザデータを格納することはできない。320、325、そして33

0はそれぞれ、レコード1のカウンタ部、キー部、データ部である。335、340そして345はそれぞれ、レコード2のカウンタ部、キー部、データ部である。以下に続くレコードは、省略した。305から345の各フィールドの間は、ギャップと呼ばれる領域で、定められた長さを持ち、データが格納されない。

【0076】380のトラック形式は、前記305から345までのフィールドを連続して配置したものである。300は、最後の有効なレコードから、バッファの終端までをうめるパディングデータであり、これについては後述する。

【0077】実際には、305から345までのフィールドにそれぞれ存在するECC(Error Correcting Code)などは380には含まれないが、図9では簡単のため、その違いは表されていない。

【0078】390、392そして394は、ディスクサブシステム80のブロック長である、512バイトの長さを持つブロックである。390、392そして394には、380に示すデータが連続して、順に格納される。CKDレコードの境界とブロックの境界とは、必ずしも一致しない。

【0079】図9に示すように、SCSIインタフェース61からは、ECCや、ギャップは見えない。

【0080】バッファ90の長さである、トラック容量は、トラックに格納されるユーザデータの最大値ではなく、各レコードのカウンタ部や、ホームアドレス、それにレコード0の長さをも加えた最大値であるとする。

【0081】図8に戻って、ステップ230では、バッファ90の、ステップ210で求めたトラック先頭からのバイト位置で指定されるブロックから、SCSIインタフェース61を経由してCPU10にデータを転送する。

【0082】ステップ240では、トラックに格納されている実際のデータ長が、トラック容量より小さいかを判断する。小さければ、ステップ250に進む。そうでない、すなわちトラック容量いっぱいのデータが格納されていれば、リード処理を終わる。トラックに格納されている実際のデータ長とは、図9に示したホームアドレスおよび各レコードのカウンタ部、キー部そしてデータ部の長さの総和をしめす。

【0083】ステップ250では、現在転送中のブロックの終端まで、0を転送する。これを以下、パディングと呼ぶ。この処理が必要な理由を以下に示す。

【0084】CKD形式では、トラックに格納できるデータの容量は、レコード長によって変わる。このため、一般的にはトラックに実際に格納されているデータの容量は、ブロック長、現在の場合512バイトの倍数にならない。ステップ250のパディングを行なわないならば、このディスクをFBA形式で読んだときに、1つのブロック内に異なるトラックのデータが混在することにな

る。これは、CPU10のCKDレコードアクセスライブラリ35の処理を複雑にする。本発明の実施の形態では、ステップ250のパディングを行ない、トラックは必ずブロック境界から始まることを保証するものとする。ステップ250のパディングを行わない構成も可能である。パディングのために転送するデータとして、0以外を使うことも可能である。

【0085】以上、図8の処理においても、図7について述べたように、トラックのキャッシュを行なう構成が可能である。

【0086】一第2の発明の実施の形態一

図10は、本発明の第2の発明の実施の形態の計算機システムの要部構成図である。

【0087】本発明の第2の発明の実施の形態の計算機システムの構成は、第1の発明の実施の形態のディスクサブシステム80のCKDディスク装置100を、FBA形式でアクセスされるFBAディスク装置100'に変更し、さらにCKD-FBAレコード形式変換R/W処理A4000を別のCKD-FBAレコード形式変換R/W処理B4010にした構成である。

【0088】CKD-FBAレコード形式変換R/W処理B4010は、チャンネルインタフェース71を経由してCKDレコードを受け取り、レコードの格納されるLBAを計算し、CKDレコードをFBAレコードとして適当な長さに分割して、レコード形式を、CKDから、FBAに変換し、そしてFBAレコードを格納する。逆の処理、つまりレコードの読み出しも同様に動作する。SCSI R/W処理4040は、SCSIインタフェース61を経由してCPU10が行うリードとライトの要求を処理する。

【0089】FBAディスク装置100'が複数ある構成も可能であり、これを図18に示す。図18については、後述する。

【0090】このFBAディスク装置100'は、ブロック長を512バイトとし、LBAによるアクセスを提供する。FBAディスク装置100'は、FBAレコードを格納し、CKDレコード110は、実際には、FBAレコード4020と4030として格納される。

【0091】ディスク装置にCKDレコード110が格納されることや、チャンネルインタフェース71を経由して、CKD形式にしたがったアクセスをCPU11に提供することは第1の発明の実施の形態と同じである。

【0092】図11は、FBAディスク装置100'に格納されるデータの配置を示したものである。380'はバッファ90に読み込まれるトラックの形式である。390、392そして394は、380'に示すデータが、SCSIインタフェース61を経由して、CPU10に転送されるときに形式を示したものである。390、392そして394は図9と同じである。

【0093】380'は、ホームアドレスや、レコード

0を含む。これは、ディスクサブシステム80が、チャンネルインタフェース71を経由してCKD形式にしたがったアクセスを受けるために、CKD形式特有の、いわゆるフォーマットライトコマンドを処理し、書き込まれたデータを保持する必要があるためである。フォーマットライトコマンドの例としては、WRITE HOME ADDRESS や、WRITE COUNT, KEY, AND DATAがある。

【0094】図12は、CPU10がディスクサブシステム80にSCSI READコマンドを発行したときに、ディスクサブシステム80のSCSI R/W処理4040が行なう、SCSI READ処理のフローチャートである。現在の例では、図12の処理は、CPU10のCKDレコードアクセスライブラリ35の要求にしたがって実行される。FBAディスク装置100'は、第1の発明の実施の形態において、CPU10とディスクサブシステム80の間でやりとりされるデータと、同じ形式のデータストリームを格納している。そのため、SCSI R/W処理4040は単純に、SCSI READコマンドに指定されたLBAに対応する、FBAディスク装置100'の位置から、ブロックの転送を開始する。そして、SCSI READコマンドに指定された数のブロックが転送されたら、SCSI READ処理は終わる。

【0095】図22は、CKD-FBAレコード形式変換R/W処理B4010がCPU11からチャンネルインタフェース71を経由して1トラック分のデータを受け取ったときに実行されるWRITE TRACK処理のフローチャートである。

【0096】ステップ3000で、CKD-FBAレコード形式変換R/W処理B4010はCPU11からチャンネルインタフェース71を経由して1トラック分のデータを受け取り、バッファ90に置く。そして、データを、図11のトラックフォーマット380'の形式に、変換する。

【0097】トラックフォーマット380'は、ホームアドレス、レコード0、レコード1とそれ以降のレコード、さらに場合によりパディングデータを含む。トラックフォーマット380'の長さはあらかじめ定められており、SCSI仕様で定められるブロック長、今の場合512バイト、の整数倍である。トラックフォーマット380'は、トラックの最後のレコードの後ろに、パディングデータを含むことがある。トラックフォーマット380'は、第1の発明の実施の形態において、CKD-FBAレコード形式変換R/W処理A4000がCKDディスク装置100からバッファ90に1トラック分のデータを読むときのフォーマットと同じである。

【0098】ステップ3010で、CKD-FBAレコード形式変換R/W処理B4010はバッファ90上のトラックフォーマット380'をもつデータを、512バイトのブロックに分割し、各ブロックを、FBAディ

スク装置100'に格納する。現在のステップで生成されるデータストリームの形式は、第1の発明の実施の形態において、CKD-FBAレコード形式変換R/W処理A4000がCPU10とディスクサブシステム80の間で、SCSIインタフェース61を経由してデータをやりとりするときに生成されるデータストリームの形式に等しい。

【0099】図18は、本発明の第2の発明の実施の形態の計算機システムのFBAディスク装置100'を、その約半分の容量を持つCKDディスク装置100"およびディスク装置2500で置き換えた計算機システムの要部構成図である。各ディスク装置へのデータの分配は、トラック単位で行なわれ、偶数トラック番号のものはディスク装置2500に、奇数トラック番号のものはCKDディスク装置100"に格納されるものとする。

【0100】—第3の発明の実施の形態—

図13は、本発明の第3の発明の実施の形態の計算機システムの要部構成図である。

【0101】本発明の第3の発明の実施の形態の計算機システムの構成は、第1の発明の実施の形態のCKDレコードアクセスライブラリ35を、少し異なる機能を持つCKDレコードアクセスライブラリ35"に置き換えた構成である。さらに、CKD-FBAレコード形式変換R/WプロセスA'4030は、第1の発明の実施の形態のCKD-FBAレコード形式変換R/WプロセスA4000と少し異なる。

【0102】本発明の実施の形態と第1の発明の実施の形態の主な違いを以下に簡単に述べる。

【0103】第1の発明の実施の形態では、CKDレコードからカウント部を取り除くのはCPU10のCKDレコードアクセスライブラリ35であったが、本発明の実施の形態ではディスクサブシステム80がこれを行なう。

【0104】これにともない、第1の発明の実施の形態では、CKDレコードアクセスライブラリ35はディスクサブシステム80からトラック単位でデータを読み込んだが、本発明の実施の形態ではブロック単位でデータを読み込む。そして、本発明の実施の形態ではSCSIインタフェース60を経由してCKDレコードアクセスライブラリ35が受け取るデータにはカウント部がついていないので、これだけではレコード境界が決まらない。しかしこれはVSAMをアクセスするうえでは問題にならない。この理由を、以下に説明する。

【0105】VSAMでは、ディスク装置に格納される物理的なレコードは、VSAMのユーザであるアプリケーションプログラムが扱うVSAMレコードとは対応しない。ディスク装置への格納すなわちCKDレコードの割当は、CIごとに行なわれる。CIを格納する各CKDレコードは、キー部を持たず、データ部の長さは同じVSAMデータセットに属するCKDレコードであればすべて等しく、VSAMにより1024バイトなど、適当な長さに決められる。ディスク

サブシステム80は、図8のステップ250のパディング処理をおこなうことで、CKD形式で格納されたデータをFBA形式でアクセスするときに、CKD形式でのトラック境界がFBA形式でのブロック境界をまたがらないことを保証する。このため、トラック先頭のLBAが与えられれば、目的とするレコードの位置は、レコード番号とレコード長から容易に求めることができる。

【0106】さらに、CKDレコードアクセスライブラリ35が受け取るデータにカウント部がついていないことは、VTOCをアクセスするうえでも問題にならない。VTOCは、キー部44バイト、データ部96バイトのレコードの集まりである。このため、上記VSAMの場合と同様に、レコード位置を求めるのは容易である。

【0107】即ち、CKDレコードアクセスライブラリ35は、その上位プログラム、今の場合VSAMアクセスライブラリ30から、現在処理対象としているトラックに、どんな形式のレコードが格納されているかを教わることで、レコード境界を見つけることが可能である。このためカウント部は必要ない。

【0108】ただし、VSAM以外のアクセス法には、任意長のCKDレコードを扱うものもあるため、本発明の実施の形態の方式は適用できないものがある。

【0109】図14は、CKDディスク装置100に格納されるデータの、トラック上の配置を示したものである。305から345は、図9と同じであり、トラック上の実際の物理的なデータの配置を示す。380"は、バッファ90に読み込まれるトラックの形式を示したものである。390"と392"は、380"に示すデータが、SCSIインタフェース61を経由して、CPU10に転送されるときに形式を示したものである。

【0110】380"のトラック形式は、前記305から345までのフィールドのうち、レコード1以下の、キー部とデータ部だけを集めたものである。

【0111】図15は、CKDレコードアクセスライブラリ35が行なう、CKDレコード読み込み処理のフローチャートである。

【0112】ステップ2200では、指定されたCCHHRを持つレコードを含むブロックのLBAを計算する。Rは、目的レコードのレコード番号である。

【0113】
$$LBA = (CC * \text{シリンダー容量} + HH * \text{トラック容量}) / \text{ブロック長} + ((R - 1) * \text{レコード長}) / \text{ブロック長}$$

最初の項が、目的レコードを含むトラックの先頭アドレスであり、2番目の項がトラック上のオフセットである。

【0114】さらに、ブロック内での、目的レコードの開始オフセットは、
$$\text{ブロック内オフセット} = ((R - 1) * \text{レコード長}) \% \text{ブロック長}$$
である。

【0115】ステップ2210では、UNIXオペレーティングシステム40のローIO機能をつかって、前記LBAから、 $((\text{レコード長} + (\text{ブロック長} - 1)) / \text{ブロック長})$ ブロックだけ、あるいは、レコードがブロックをまたがる場合にはさらに1ブロック余分に、SCSIインタフェース60を経由してディスクサブシステム80からデータを読む。

【0116】ステップ2220では、読み込んだデータのアドレス+ブロック内オフセットをキー部として、そしてキー部の終了アドレスをデータ部としてコール元に返して処理を終了する。

【0117】図14で、CKDレコード読み込み処理を例を使って説明する。

【0118】CKDレコードアクセスライブラリ35は、VSAMアクセスライブラリ30から、VTOCのレコード4を読む要求を受けたとする。VTOCは、トラック先頭のレコード1から開始するとする。ブロック長は512バイト、レコード長はキー部が44バイト、データ部が96バイト、合計140バイトである。

【0119】 $\text{LBA} = (\text{CC} * \text{シリンダー容量} + \text{HH} * \text{トラック容量}) / 512 + ((4-1) * 140) / 512$ であり、ブロック内オフセット $= ((4-1) * 140) \% 512 = 420$ である。

【0120】ステップ2210では、 $(140 + (512-1)) / 512 + 1 = 2$ ブロックが読み込まれる。

【0121】ステップ2220では、読んだデータの先頭から420バイト目のアドレスがキー部として、464バイト目のアドレスがデータ部として、VSAMアクセスライブラリ30に戻る。

【0122】図23は、CPU10がディスクサブシステム80にSCSI READコマンドを発行したときに、ディスクサブシステム80のCKD-FBAレコード形式変換R/WプロセスA'4030によって実行される、SCSI READ 処理のフローチャートである。CKD-FBAレコード形式変換R/WプロセスA'4000の行うSCSI READ処理のフローチャートとの違いは、ステップ3100で、CKD-FBAレコード形式変換R/WプロセスA'4030はバッファ90に読まれたCKDレコードから、キー部とデータ部を抜き出し、これらだけを、SCSIインタフェース61を経由してCPU10に転送することである。SCSI READ 処理のそのほかのステップは、発明の第1の発明の実施の形態と同じであるので、説明は省略する。

【0123】一第4の発明の実施の形態—

図16は、本発明の第4の発明の実施の形態の計算機システムの要部構成図である。

【0124】本発明の第4の発明の実施の形態の計算機システムの構成は、第1の発明の実施の形態のVSAMアクセスライブラリ30を、少し異なる機能を持つVS

AMアクセスライブラリ30' ' 'に置き換え、さらにCPU11にVTOCユティリティ2300を加えた構成である。

【0125】本発明の実施の形態と第1の発明の実施の形態の主な違いを以下に簡単に述べる。

【0126】第1の発明の実施の形態のVSAMアクセスライブラリ30は、図4に示したVSAM OPEN処理において、指定されたデータセット名称の形式1DSCBを探すために、標準ボリュームラベルとVTOCを読んだ。本発明の実施の形態のVSAMアクセスライブラリ30' ' 'は、そのためにVTOCユティリティ2300が求めたデータを使い、ディスクサブシステム80のアクセスはしない。

【0127】VTOCユティリティ2300は、VOS3オペレーティングシステム50上で動作し、ディスクサブシステム80をアクセスして目的データセットの形式1DSCBを求める。VOS3オペレーティングシステム50にはこれを行なうマクロが提供されており、それをOBTAIN(SEARCH)マクロと言う。VTOCユティリティ2300は、OBTAIN(SEARCH)マクロを、目的とするデータセット名称を指定して発行する。VOS3オペレーティングシステム50はVTOCをアクセスして、結果である形式1DSCBをVTOCユティリティ2300に返す。

【0128】VOS3オペレーティングシステム50はVTOCをアクセスするときに、VTOC索引を使用したり、指定したキー部を持つレコードをサーチするCCW(Channel Command Word)を使用したりするために、一般的にその処理は第1の発明の実施の形態のVSAMアクセスライブラリ30の行なう、図4に示したVSAM OPEN処理のステップ505からステップ540に比べると高速である。

【0129】図17は、VSAMアクセスライブラリ30' ' 'が行う、VSAM ESDSのオープン処理のフローチャートである。

【0130】ステップ500で、UNIXオペレーティングシステム40の、ローIO機能をオープンする。

【0131】ステップ2400では、VSAMアクセスライブラリ30' ' 'はVTOCユティリティ2300より形式1DSCBを得る。具体的には、プログラム間通信を用いてVTOCユティリティ2300を呼び出して結果を返送させてもいいし、あらかじめVTOCユティリティ2300を実行させておき、その結果を人間がコンソールから入力してもよい。

【0132】ステップ550では、形式1DSCBをVSAMアクセスライブラリ30のローカルな変数にコピーして、コール元に戻る。これで、VSAM ESDSのオープン処理が終わる。

【0133】一第5の発明の実施の形態—

図19は、本発明の第5の発明の実施の形態の計算機システムの要部構成図である。

【0134】本発明の第5の発明の実施の形態の計算機

システムの構成は、第1の発明の実施の形態のVSAMアクセスライブラリ30を、少し異なる機能を持つVSAMアクセスライブラリ30"に置き換え、さらにCPU11にロックユティリティ2600を加えた構成である。

【0135】本発明の実施の形態は、第1の発明の実施の形態が有する、以下の課題を解決するための機構を有する。

【0136】VOS3オペレーティングシステム50はカウントキーデータ形式に従うインタフェースでデータをアクセスし、当該データを管理し、UNIXオペレーティングシステム40は固定長ブロック形式に従うインタフェースでデータをアクセスし、当該データを管理する。両オペレーティングシステムは、CKDディスク装置100にある各データセットが使用中かどうかという情報を、それぞれのオペレーティングシステムの管理するメモリに持つ。このため、お互いがどのデータセットを使用しているかはわからない。このため、UNIXオペレーティングシステム40上で動作するアプリケーションプログラム20が現在参照しているデータセットを、VOS3オペレーティングシステム50上で動作するアプリケーションプログラム21が削除してしまうといった不具合を防ぐことができない。このような不具合は、アプリケーションプログラムあるいはオペレーティングシステムの動作不良あるいはデータセットの破壊をひきおこすおそれがあるために、避けなければいけない。このために、本発明の実施の形態はロックユティリティ2600を有する。

【0137】ロックユティリティ2600は、VOS3オペレーティングシステム50上で動作するプログラムである。ロックユティリティ2600は、VSAMアクセスライブラリ30"の要求に従って、VSAMアクセスライブラリ30"がアクセスしようとするデータセットの使用を、VOS3オペレーティングシステム50に要求する。しかしロックユティリティ2600は自分では使用を要求したデータセットへのアクセスは行わない。データセットへのアクセスはSCSIインタフェースを経由してUNIXオペレーティングシステム40が行う。VOS3オペレーティングシステム50には必要に応じてデータセットの使用の要求を行なうマクロが提供されており、それをDYNALLOCマクロと言う。VSAMアクセスライブラリ30"は、データセットをアクセスしようとするときには、そのデータセット名称を指定してロックユティリティ2600を呼ぶ。ロックユティリティ2600は、DYNALLOCマクロを、目的とするデータセット名称を指定して発行する。これにより、UNIXオペレーティングシステム40がアクセスしようとするデータセットは、VOS3オペレーティングシステム50にとっても使用中とみなされるため、前述のアクセスの競合による不具合を防ぐことができる。

【0138】図20は、VSAMアクセスライブラリ30"が行う、VSAM ESDSのオープン処理のフローチャートである。

【0139】ステップ2700で、VSAMアクセスライブラリ30"はオープンしようとするデータセット名称を指定して、ロックユティリティ2600を呼ぶ。具体的には、プログラム間通信を用いてロックユティリティ2600を呼び出すなどの方法がある。ロックユティリティ2600はVOS3オペレーティングシステム50にDYNALLOCマクロを発行し、データセットの使用を要求する。これにより、当該データセットは、VOS3オペレーティングシステム50で動作するアプリケーションプログラムからアクセスができなくなる。

【0140】以下の処理は図4と同じなので、説明は省略する。

【0141】図21は、VSAMアクセスライブラリ30"が行う、VSAM ESDSのクローズ処理のフローチャートである。

【0142】ステップ2800で、VSAMアクセスライブラリ30"はクローズしようとするデータセット名称を指定して、ロックユティリティ2600を呼ぶ。ロックユティリティ2600はVOS3オペレーティングシステム50にDYNALLOCマクロを発行し、データセットの使用の終了を伝える。これにより、当該データセットは、VOS3オペレーティングシステム50で動作するアプリケーションプログラムからアクセスできるようになる。

【0143】以下の処理は図6と同じなので、説明は省略する。

【0144】本発明の実施の形態では、VSAMアクセスライブラリ30"がオープンしたデータセットは、VOS3オペレーティングシステム50で動作するアプリケーションプログラムからはいっさいアクセスできなくなる。これを、例えばデータセットの読みだしは許可し、更新は許可しないようにすることもできる。そのためには、VSAMアクセスライブラリ30"からロックユティリティ2600に、データセット名称の他に当該データセットの共用モードを与え、ロックユティリティ2600がその共用モードに従ってDYNALLOCマクロを発行するようにすればよい。

【0145】また、VSAMアクセスライブラリ30"はロックユティリティ2600を使用して、UNIXオペレーティングシステム40上で動作するアプリケーションプログラム20の要求にしたがって、CKDディスク装置100にVOS3オペレーティングシステム50の管理するデータセットを新規に作成することができる。同様に、データセットを削除したり、既存のデータセットの大きさを拡張、縮小することもできる。

【0146】

【発明の効果】本発明の計算機システムによれば、メイ

ンフレームと、オープンシステムとの間で、データが共用できる。これにより、より柔軟で、安価かつ高性能な計算機システムを構成することができる。

【図面の簡単な説明】

【図1】本発明の第1の発明の実施の形態の計算機システムの要部構成図である。

【図2】本発明の第1の発明の実施の形態のUNIXオペレーティングシステムの要部構成図である。

【図3】本発明の第1の発明の実施の形態のアプリケーションプログラムの処理の手順を示すフローチャートである。

【図4】本発明の第1の発明の実施の形態のVSAM OPEN処理の手順を示すフローチャートである。

【図5】本発明の第1の発明の実施の形態のVSAM READ処理の手順を示すフローチャートである。

【図6】本発明の第1の発明の実施の形態のVSAM CLOSE処理の手順を示すフローチャートである。

【図7】本発明の第1の発明の実施の形態のCKDレコード読み込み処理の手順を示すフローチャートである。

【図8】本発明の第1の発明の実施の形態のSCSI READ処理の手順を示すフローチャートである。

【図9】本発明の第1の発明の実施の形態のトラック形式のデータ構造図である。

【図10】本発明の第2の発明の実施の形態の計算機システムの要部構成図である。

【図11】本発明の第2の発明の実施の形態のトラック形式のデータ構造図である。

【図12】本発明の第2の発明の実施の形態のSCSI READ処理の手順を示すフローチャートである。

【図13】本発明の第3の発明の実施の形態の計算機システムの要部構成図である。

【図14】本発明の第3の発明の実施の形態のトラック形式のデータ構造図である。

【図15】本発明の第3の発明の実施の形態のCKDレコード読み込み処理の手順を示すフローチャートである。

【図16】本発明の第4の発明の実施の形態の計算機シ

ステムの要部構成図である。

【図17】本発明の第4の発明の実施の形態のVSAM OPEN処理の手順を示すフローチャートである。

【図18】本発明の第2の発明の実施の形態の計算機システムの要部構成図である。

【図19】本発明の第5の発明の実施の形態の計算機システムの要部構成図である。

【図20】本発明の第5の発明の実施の形態のVSAM OPEN処理の手順を示すフローチャートである。

【図21】本発明の第5の発明の実施の形態のVSAM CLOSE処理の手順を示すフローチャートである。

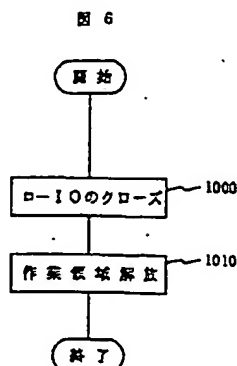
【図22】本発明の第2の発明の実施の形態のWRITE TRACK 処理の手順を示すフローチャートである。

【図23】本発明の第3の発明の実施の形態のSCSI READ 処理の手順を示すフローチャートである。

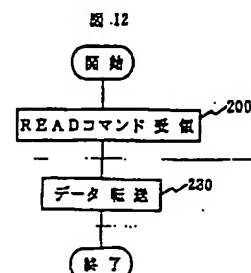
【符号の説明】

10、11：CPU、
20、21：アプリケーションプログラム、
30、30'、30''：VSAMアクセスライブラリ、
35、35'：CKDレコードアクセスライブラリ、
40：UNIXオペレーティングシステム、
50：VOS3オペレーティングシステム、
60、61：SCSIインタフェース、
70、71：チャネルインタフェース、
80：ディスクサブシステム、
90：バッファ、
100、100'：CKDディスク装置、
100'：FBAディスク装置、
4000：CKD-FBAレコード形式変換R/W処理A、
4010：CKD-FBAレコード形式変換R/W処理B、
4030：CKD-FBAレコード形式変換R/W処理A'、
4040：SCSI R/W処理

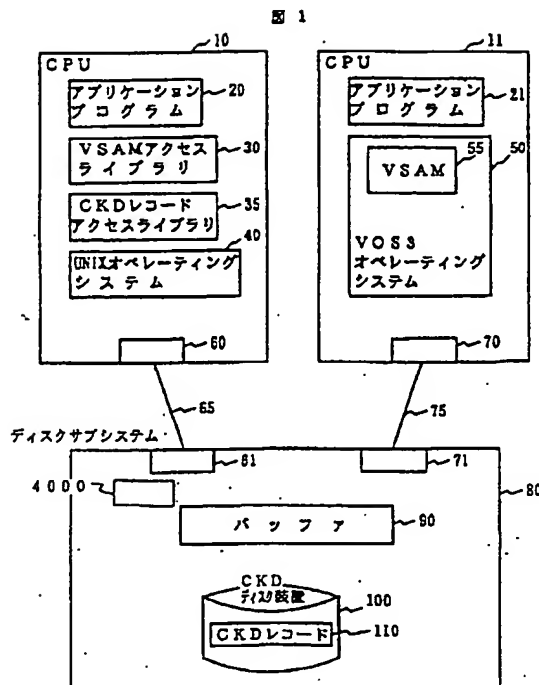
【図6】



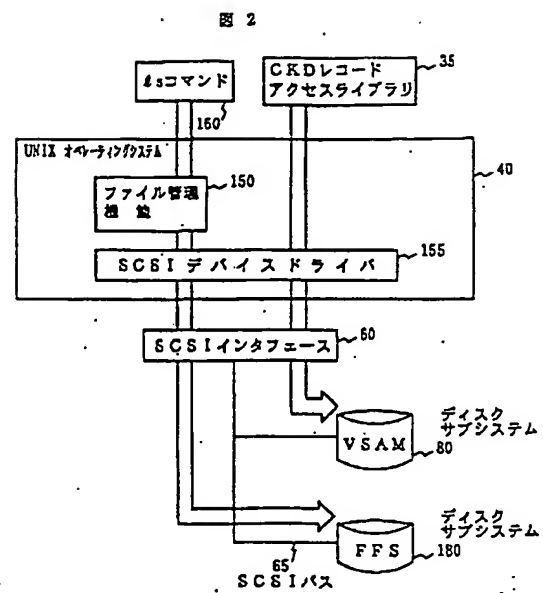
【図12】



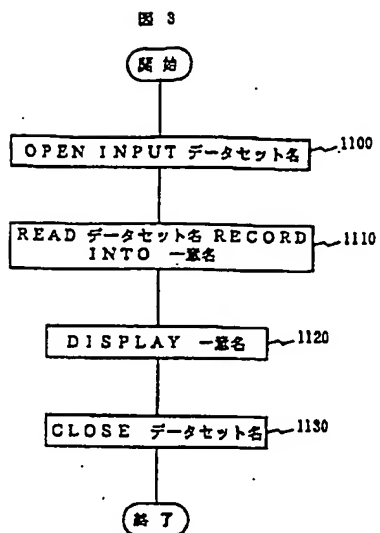
【図1】



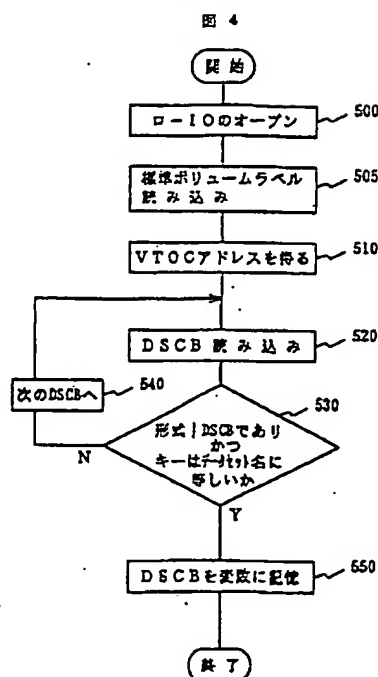
【図2】



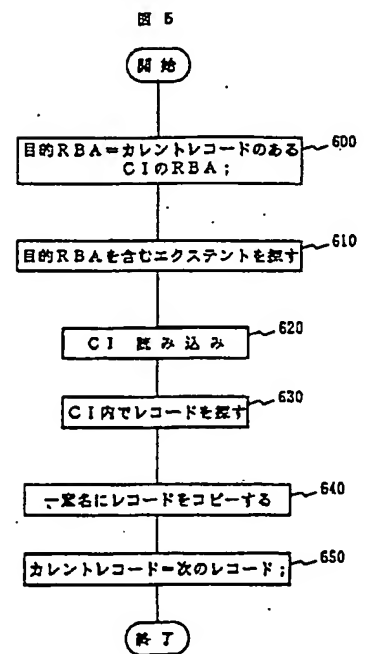
【図3】



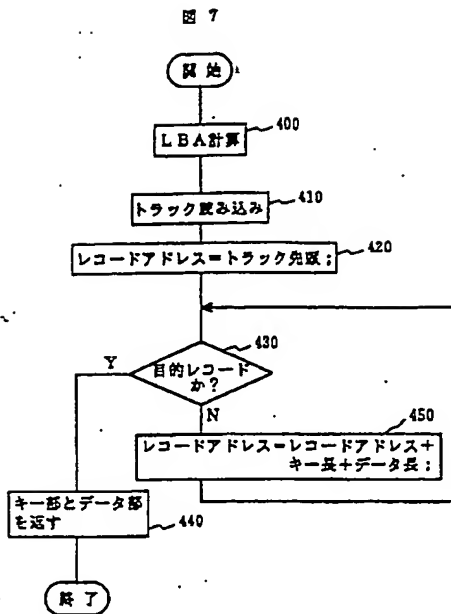
【図4】



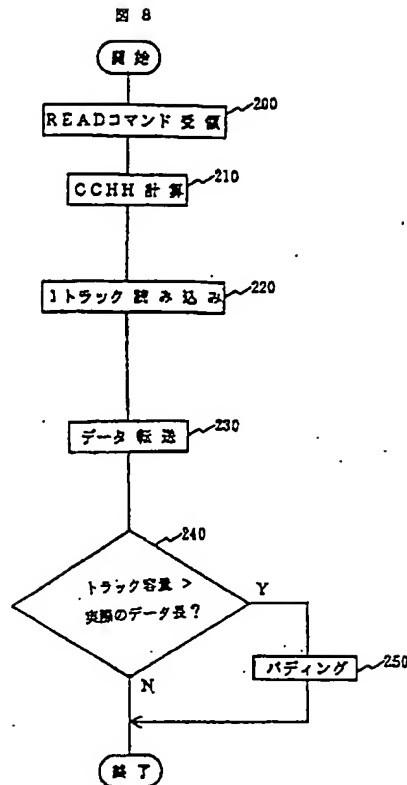
【図5】



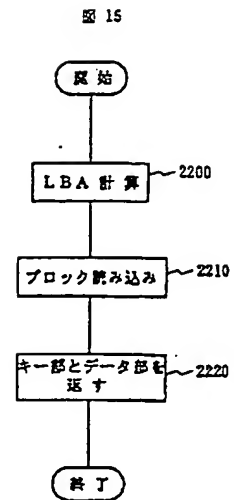
【図7】



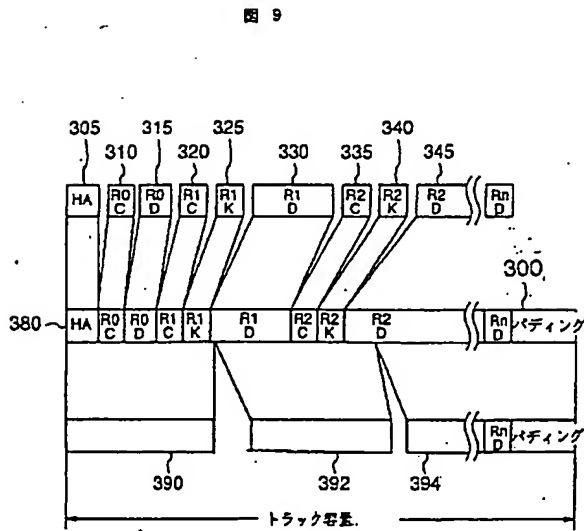
【図8】



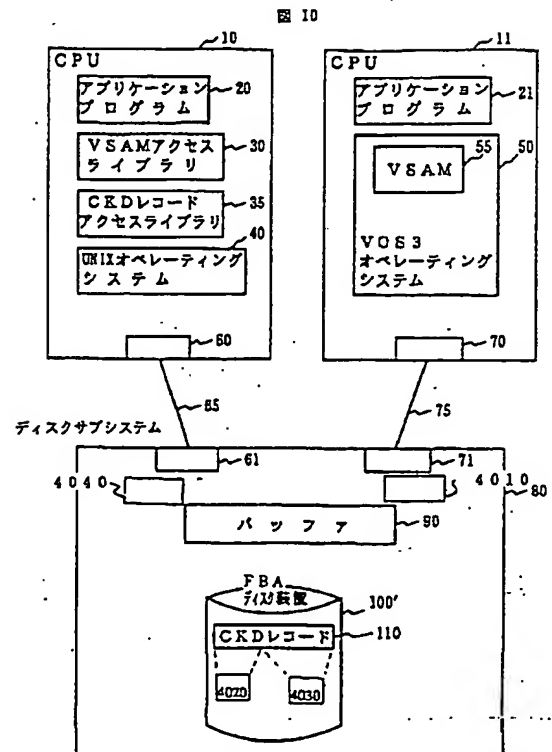
【図15】



【図9】

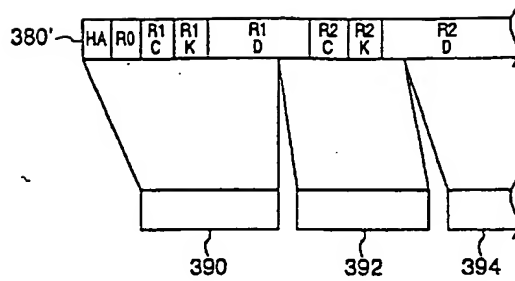


【図10】



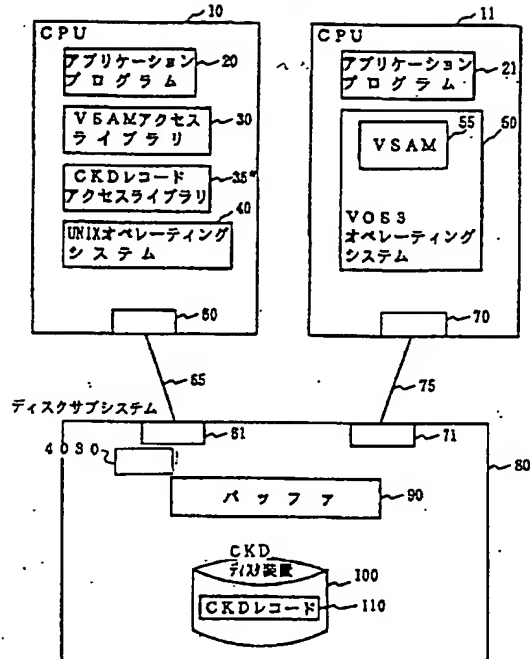
【図11】

図 11



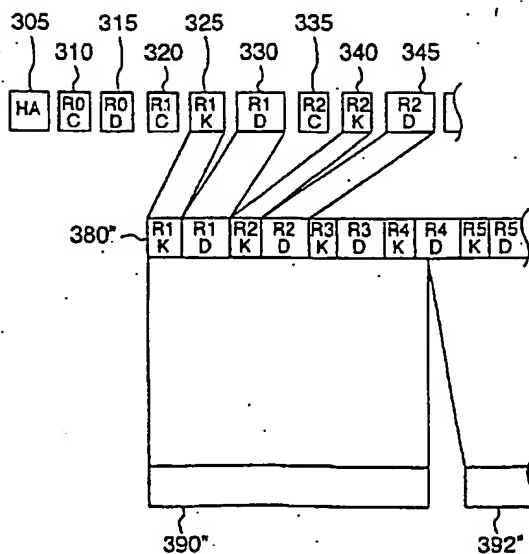
【図13】

図 13



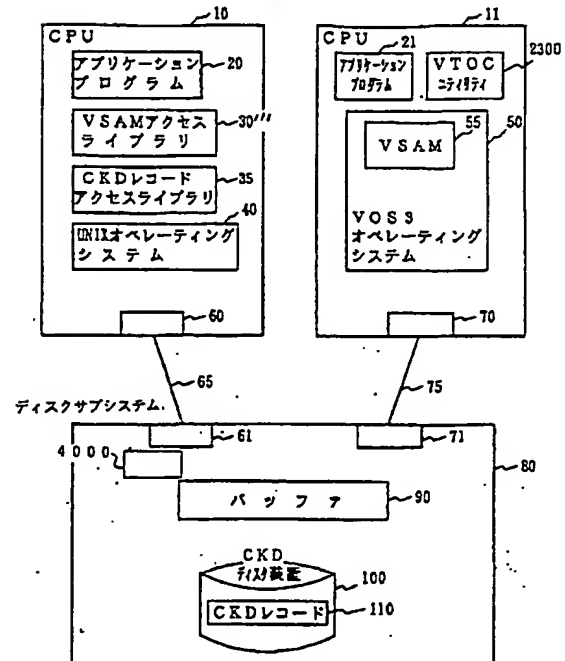
【図14】

図 14

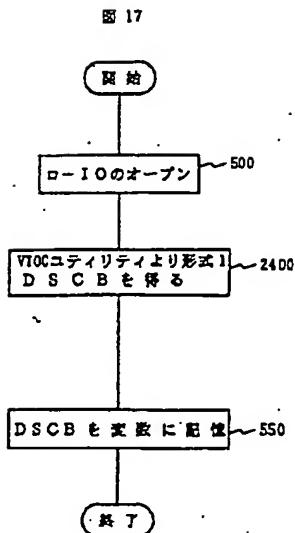


【図16】

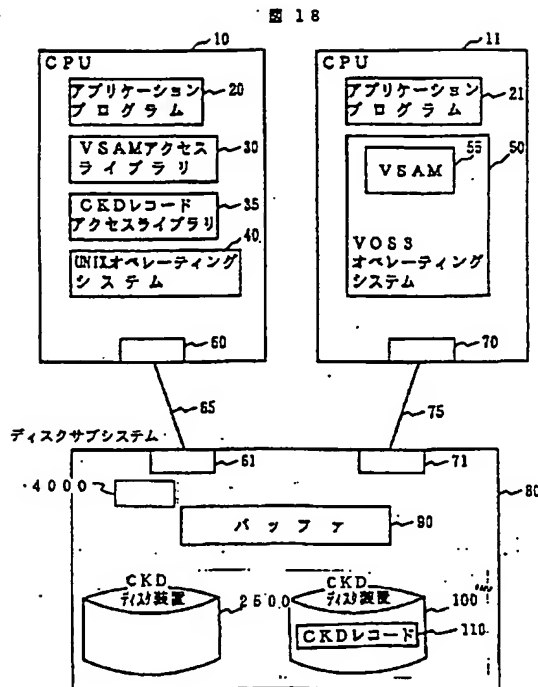
図 16



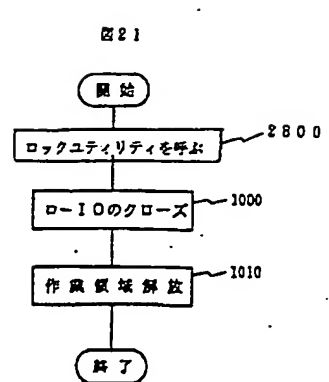
【図17】



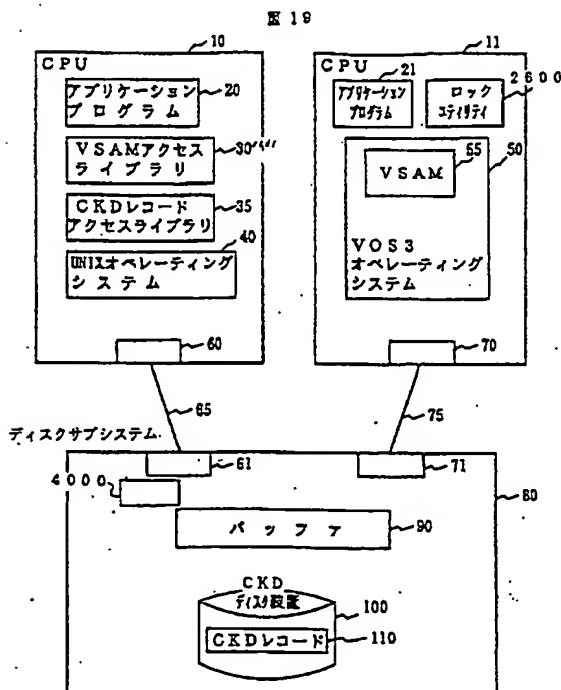
【図18】



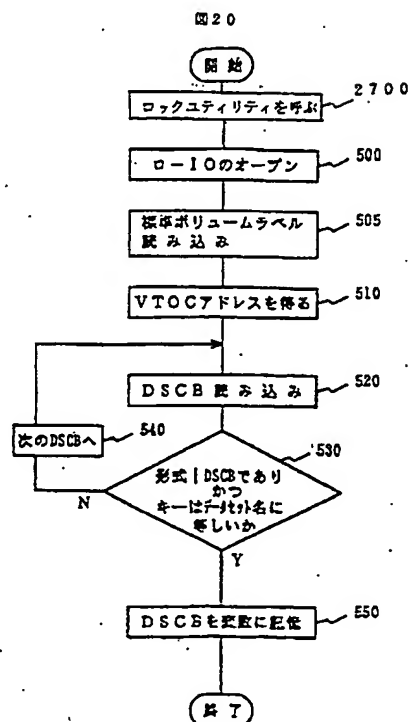
【図21】



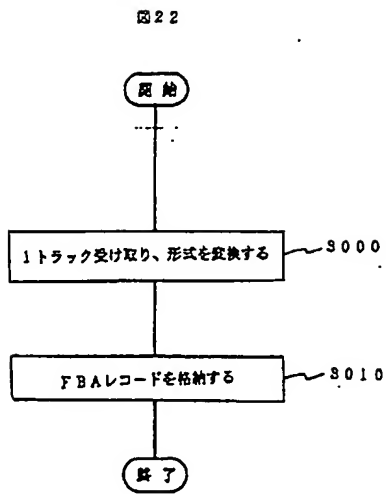
【図19】



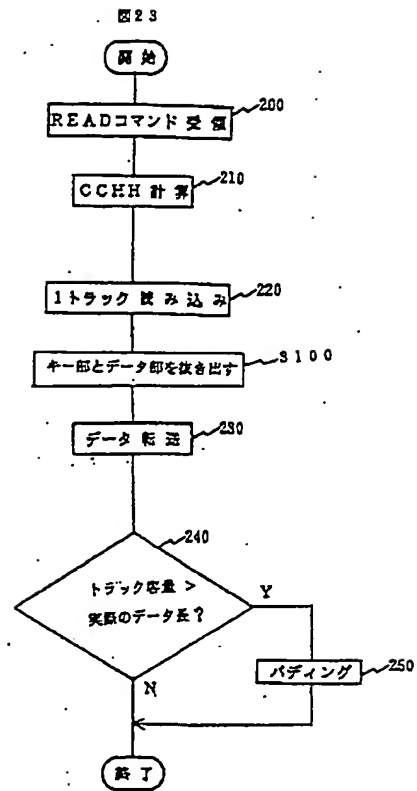
【図20】



【図22】



【図23】

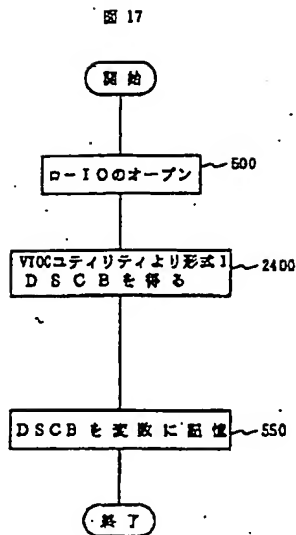


フロントページの続き

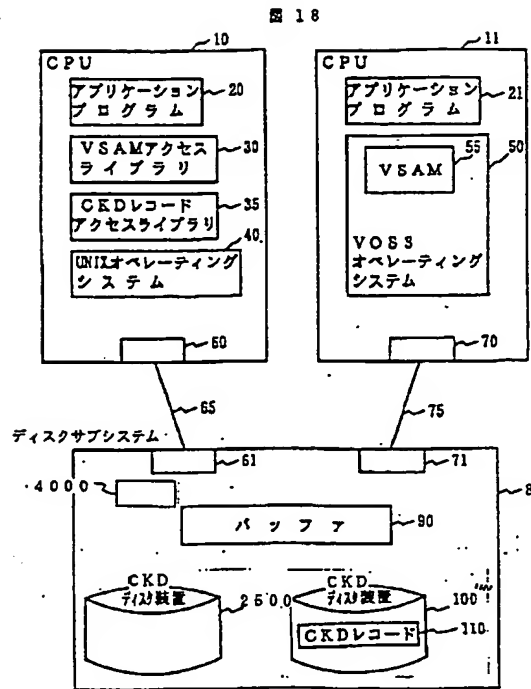
(72)発明者 吉田 稔

神奈川県小田原市国府津2880番地 株式会
社日立製作所ストレージシステム事業部内

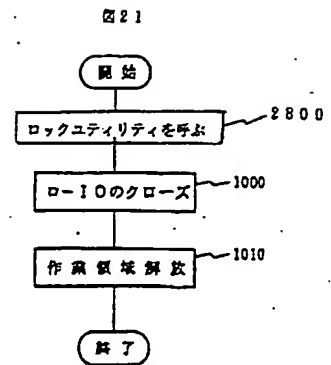
【図17】



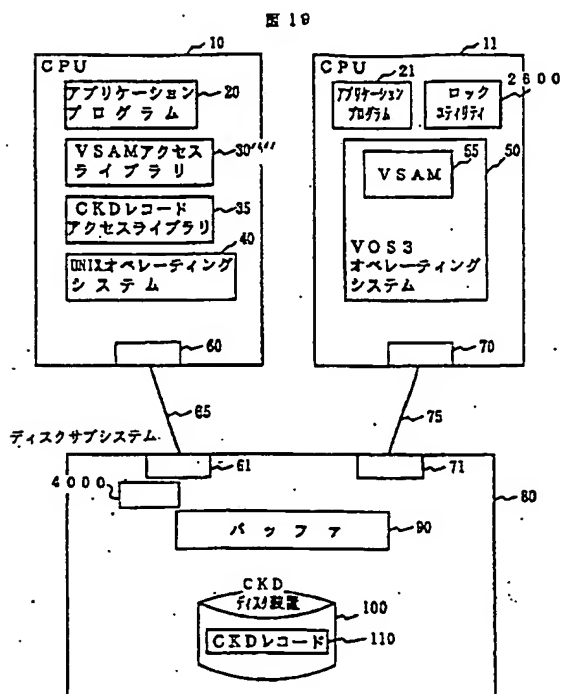
【図18】



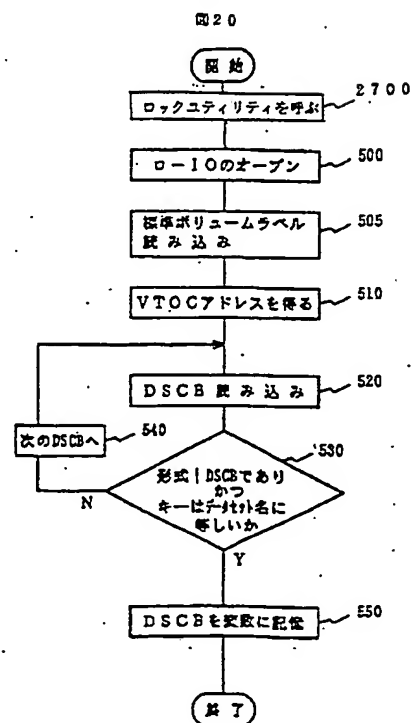
【図21】



【図19】



【図20】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.